

DEPLOYMENT OF APP IN IBM CLOUD

Team ID	PNT2022TMID04798
Project Name	CONTAINMENT ZONE ALERTING APPLICATION

DEPLOY IN KUBERNETES CLUSTER

1. Target the IBM Cloud Kubernetes Service region where you want to work. `ibmcloud cs region-set us-south`
2. Set the context for the cluster in your CLI.
 - a. Get the command to set the environment variable and download the Kubernetes configuration files. `ibmcloud cs cluster-config cluster_kunal`
 - b. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal. The command output should look similar to the following.

```
> export
```

```
KUBECONFIG=/Users/$USER/.bluemix/plugins/container-service/clusters/<  
cluster_name >/< cluster_configuration_file.yaml>
```

3. Verify that you can connect to your cluster by listing your workernodes.

```
kubectl get nodes
```

4. Create the deployment.

```
kubectl create -f deployment.yaml
```

5. Create the service.

```
kubectl create -f service.yaml
```

6. Look at the Kubernetes dashboard from the IBM KubernetesService overview page.

kubernetes

Search

+ CREATE

Overview

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Ingresses

Services

Config and Storage

Config Maps

Deployments

Name	Labels	Pods	Age	Images
flask-node-deployment	app: flasknode	1 / 1	5 minutes	registry.ng.bluemix.net/flask-node/app

Pods

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
flask-node-deployment-5cd96cf6bc-d6nfx	10.47.79.201	Running	0	5 minutes	0	19.352 Mi

Replica Sets

Name	Labels	Pods	Age	Images
flask-node-deployment-5cd96cf6bc	app: flasknode pod-template-hash: 1785279267	1 / 1	5 minutes	registry.ng.bluemix.net/flask-node/app

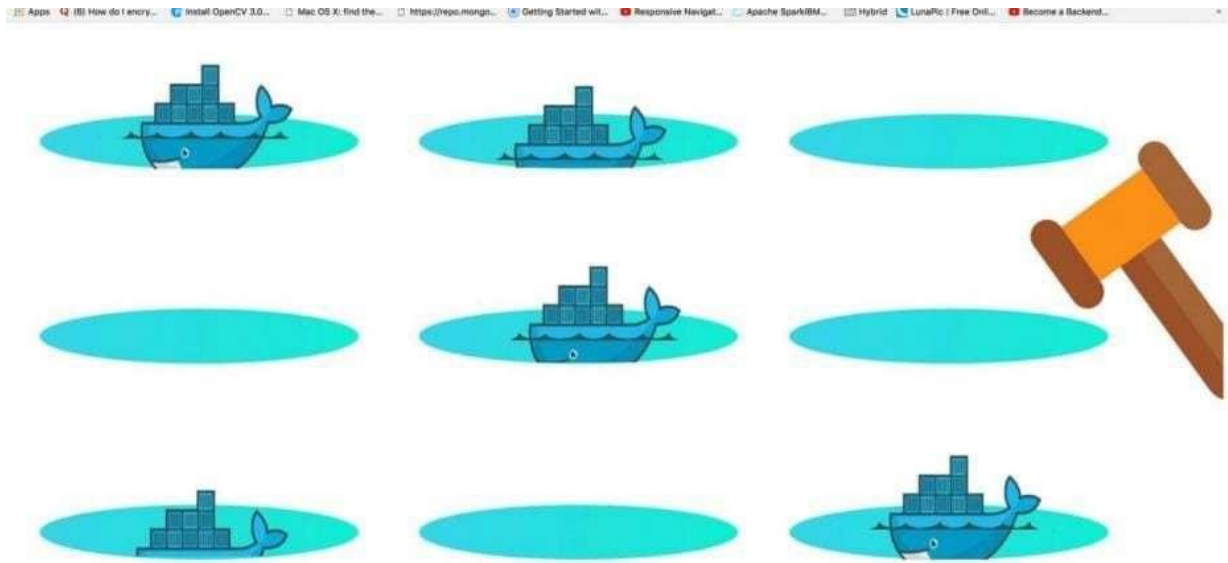
Discovery and Load Balancing

Services

Name	Labels	Cluster IP	Internal endpoints	External endpoints	Age
kubernetes	component: apiserver provider: kubernetes	172.21.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	a minute
flask-node-deployment	-	172.21.104.14	flask-node-deployment:5000 TCP flask-node-deployment:0 TCP	-	a minute

Config and Storage

7. Finally, go to your browser and ping the Public IP of your worker node



SAMPLE:

...

```
$ kubectl apply -f ./deploys/k8s-deployment/deployment.yaml
deployment.apps/k8s-nginx-deployment created
```

...

RUN THE APPLICATION

...

```
$ kubectl port-forward svc/k8s-nginx-deployment 8080:80
```

...

VERIFY THE APPLICATION RUNNING IN THE CLUSTER

...

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
k8s-nginx-deployment-68d8f8f8f4-f4x6x	1/1	Running	0	30s

...

...

```
$ curl localhost:8080
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
Welcome to nginx!  
<style> body
```

```
{
```

```
width: 35em; margin:
```

```
0 auto;
```

```
font-family: Tahoma, Verdana, Arial
```